**EXPERIMENT NO- 3**

**AIM:** WAP to find factorial of number using procedure.

**Resource Required:** P-IV and above RAM 128MB, Dot Matrix Printer, Emu 8086, MASM

611/ TASM, Turbo C/C++, Printer, Printout Stationary.

**THEORY:**

Instructions used in this program are:

**1) JNZ (Jump if not Zero):** This is conditional Jump. This instruction will jump to

specified label when zero flag is not set.

**2) DEC:** DEC decrements the source by one

***Syntax***: DEC source

**3) CMP:** Compare the numerical value of the destination with the source and set flags

appropriately. This comparison is carried out in the form of a subtraction to determine

which of the operands has a greater value. After a CMP instruction, OF, SF, ZF and CF

are set appropriately. For example, if the operands have equal values, then ZF if set.

***Syntax:***

CMP destination, source

**4) CALL AND RET :** These instructions interrupts the flow of a program by passing

control to an internal or external subroutine .The return instruction returns the control

from a subroutine back to a calling program .CALL passes the control to a label

specified after the call keyword. When the subroutine ends with return instruction, the

instruction following CALL are processed.

**ALGORITHM:**

Step I : Initialize the data segment

Step II : Initialize the variable A to number

Step III : Move the contents of variable A to AX

Step IV : Call procedure factorial

Step V : Decrement the value of variable A

Step VI : Multiply A with AX

Step VII : Copy the value of A to CX

Step VIII : Compare contents of CX with 01

Step IX : If equal go to step X else step V

Step X : Return to calling program

Step XI : Display the value in Fact variable

Step XII : Stop

**CONCLUSION: We have successfully calculated Factorial of desired number using Procedural as well as normal programming approach in Assembly language using EMU 8086.**

**Procedural Code:**

**Data segment**

**A dw 0007h**

**fact dw ?**

**Data ends**

**Code segment**

**assume cs:Code ds:Data**

**Start:**

**mov ax,Data**

**mov ds,ax**

**mov ax,a**

**call factorial**

**mov fact,ax**

**int 3h**

**factorial proc**

**label:**

**dec a**

**mul a**

**mov cx,a**

**cmp cx,0001h**

**jnz label**

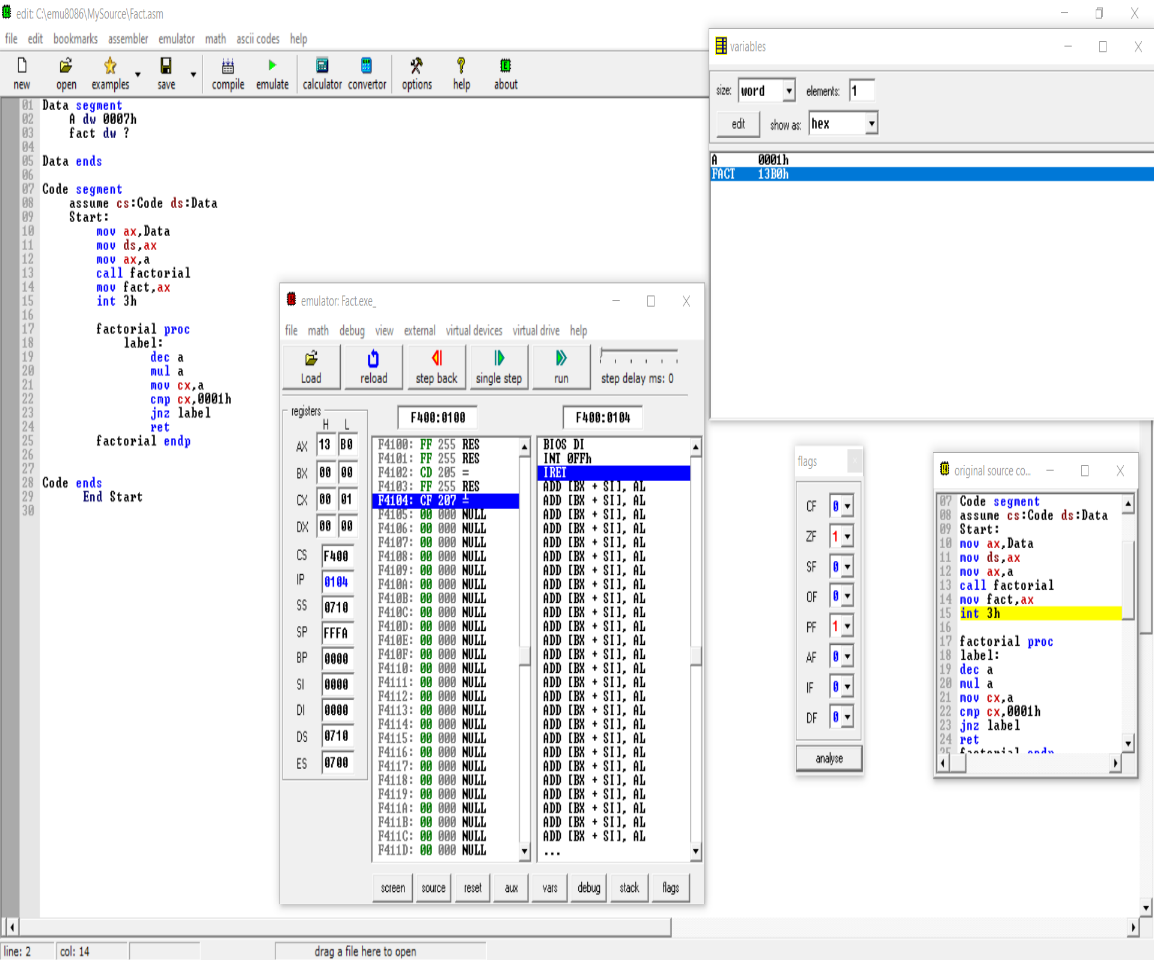
**ret**

**factorial endp**

**Code ends**

**End Start**

**Output:**

****

**Normal code:**

**Data segment**

**A dw 0007h**

**fact dw ?**

**Data ends**

**Code segment**

**assume cs:Code ds:Data**

**Start:**

**mov ax,Data**

**mov ds,ax**

**mov ax,a**

**label:**

**dec a**

**mul a**

**mov cx,a**

**cmp cx,0001h**

**jnz label**

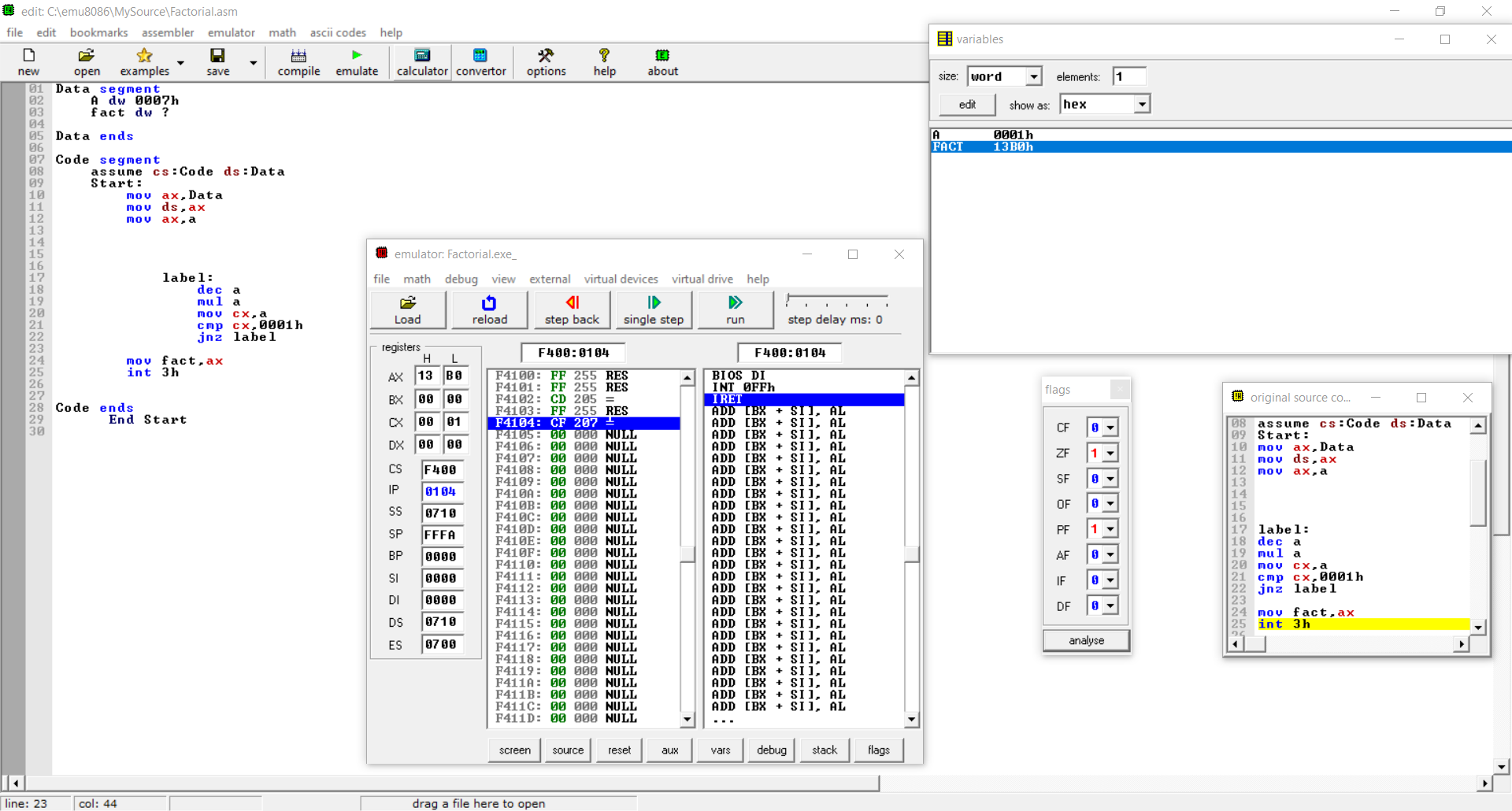
**mov fact,ax**

**int 3h**

**Code ends**

**End Start**

**Output:**

****